

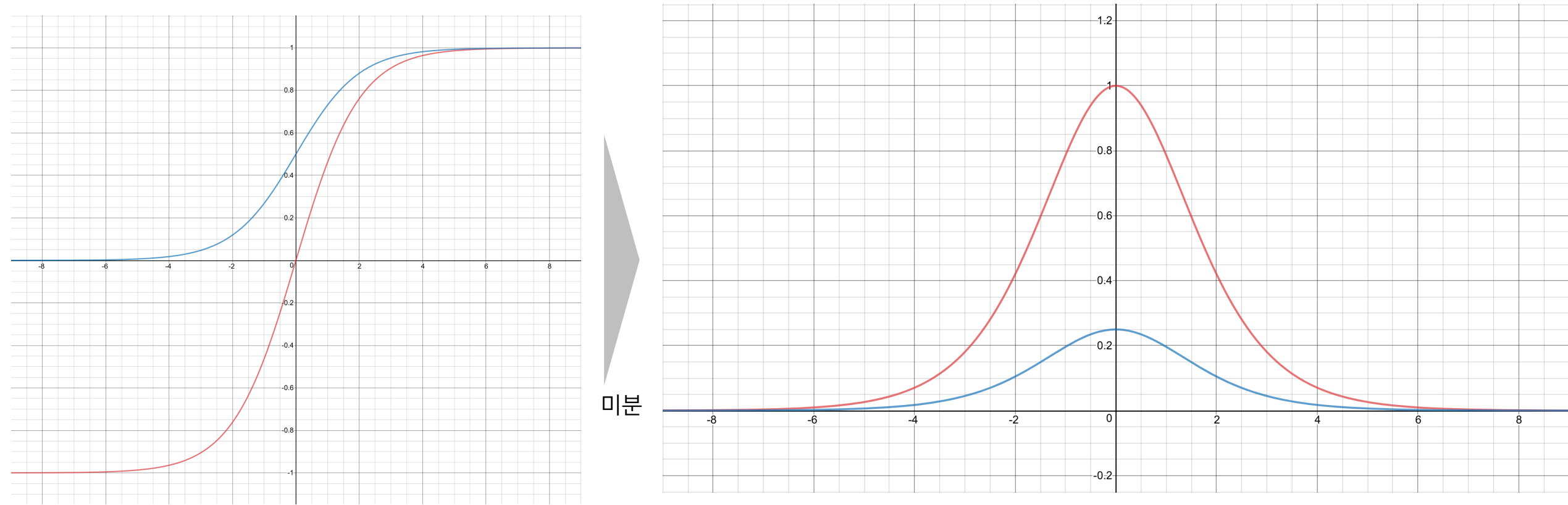
# ReLU

Ki Hyun Kim

[nlp.with.deep.learning@gmail.com](mailto:nlp.with.deep.learning@gmail.com)

# Gradient Vanishing by Sigmoid & TanH

- Chain rule로 펼쳤을 때, 이 함수에 대한 미분 값은 항상 1보다 같거나 작다.



# Gradient Vanishing by Sigmoid & TanH

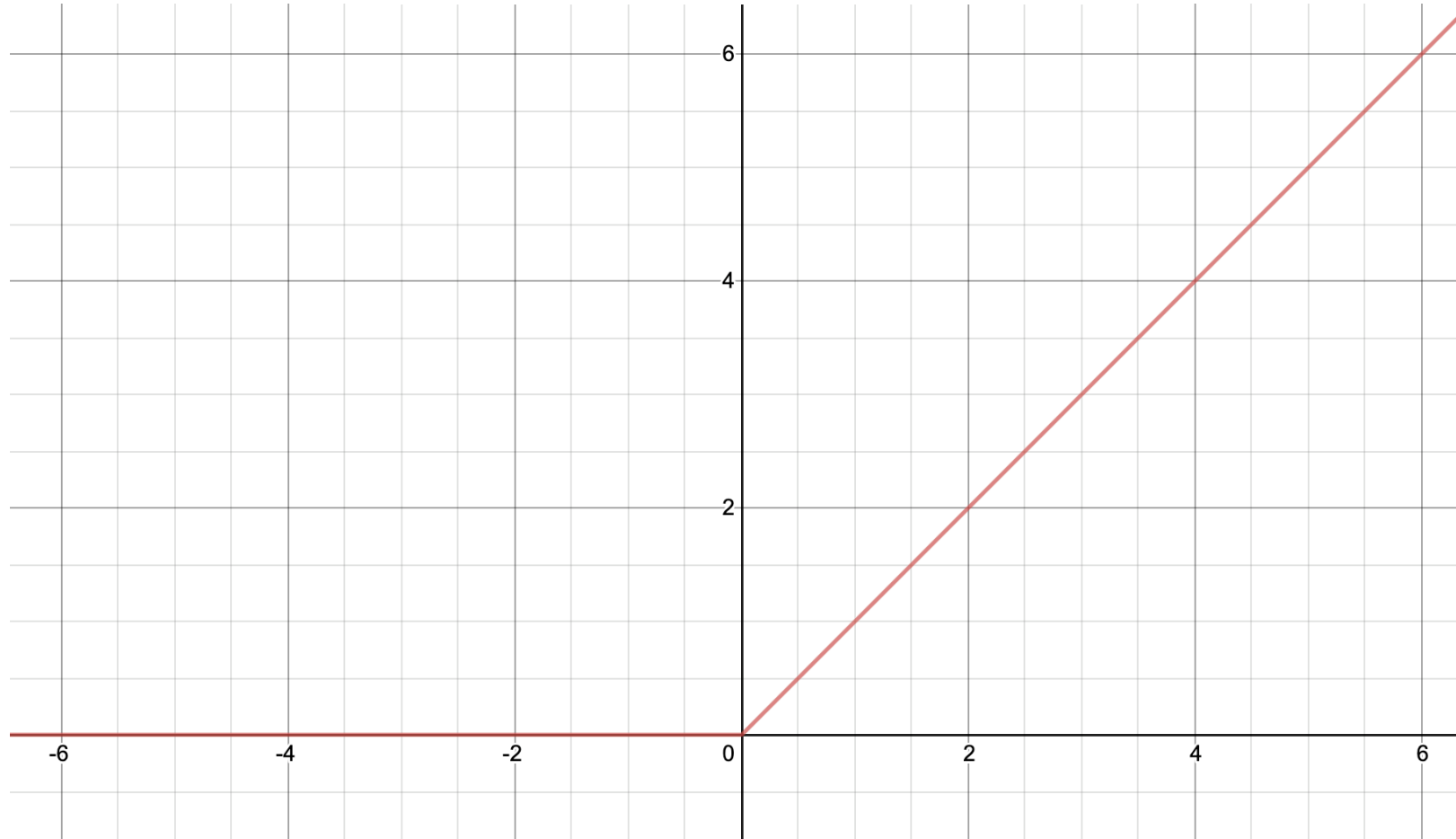
- DNN이 깊어지게 되면 활성화 함수의 사용 횟수가 증가 할 것
  - 따라서 입력에 가까운 파라미터에 대한 미분에선 1보다 작은 값이 반복적으로 곱해질 것

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_1} &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W_1} \\ &= \frac{\partial \mathcal{L}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} \cdot \frac{\partial h_2}{\partial \tilde{h}_2} \cdot \frac{\partial \tilde{h}_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial \tilde{h}_1} \cdot \frac{\partial \tilde{h}_1}{\partial W_1} \\ &\quad \text{where } \frac{\partial h_\ell}{\partial \tilde{h}_\ell} = \frac{\partial \sigma}{\partial \tilde{h}_\ell} < 1.\end{aligned}$$

# ReLU (Rectified Linear Unit)

- 두 개의 linear 함수로 이루어져 있음

$$y = \text{ReLU}(x) = \max(0, x)$$

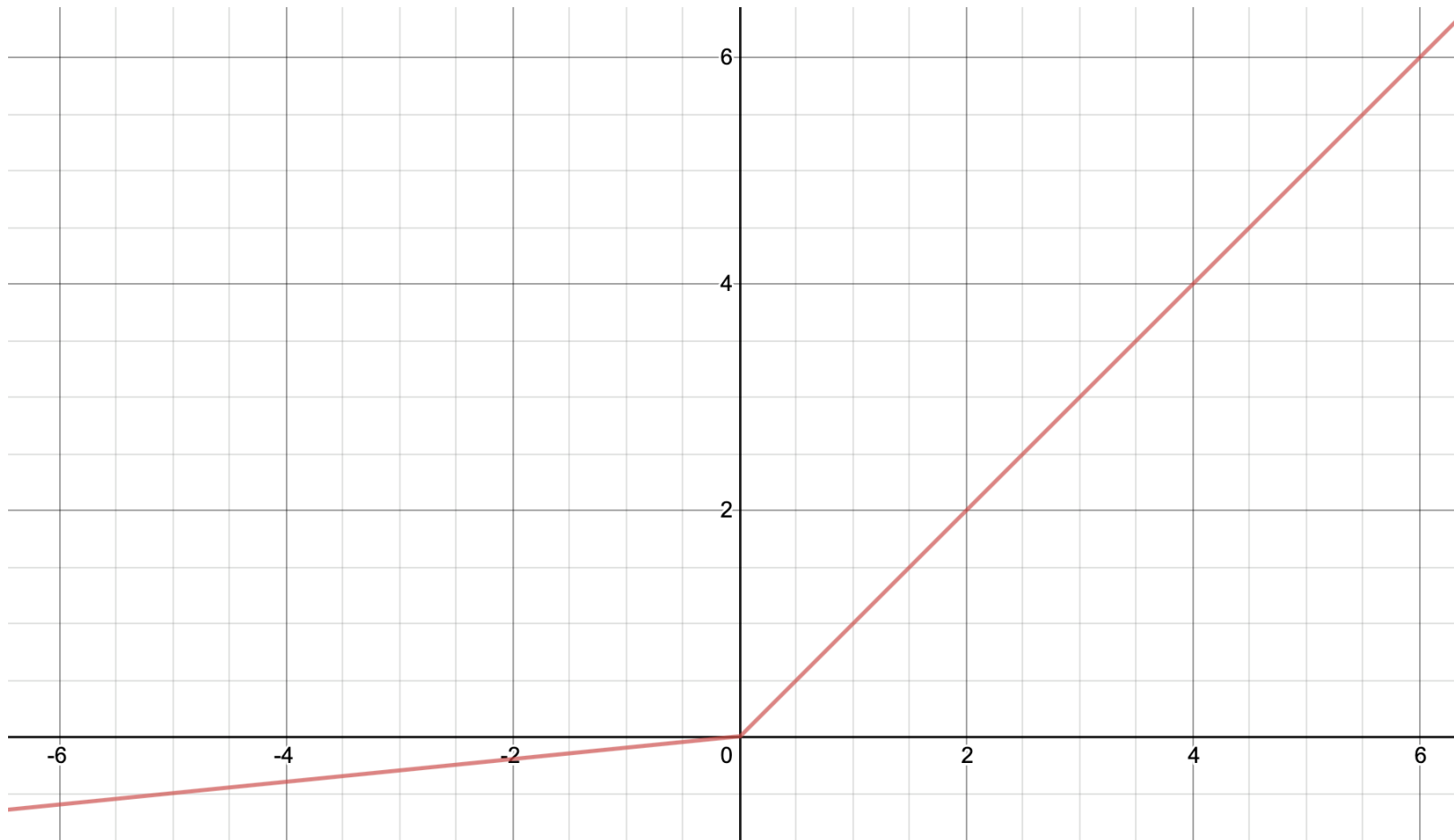


# Leaky ReLU

- 기울기를 조절할 수 있음

$$y = \text{LeakyReLU}_{\alpha}(x) = \max(\alpha \cdot x, x),$$

where  $0 \leq \alpha < 1$ .



# Summary

- 기존의 Sigmoid, TanH는 gradient vanishing 문제를 일으킴
  - 함수의 미분 값이 항상 1보다 작거나 같음
- ReLU를 통해 gradient vanishing 문제를 어느정도 해결할 수 있다.
- 양수 부분의 기울기는 항상 1이므로 학습 속도가 빠르다.
  - 또한 linear에 가까운 특성으로 인해, 최적화가 더 쉽다.
- ReLU의 입력 값이 음수인 경우, 이전 레이어는 학습이 불가하다.
  - LeakyReLU를 통해 단점을 극복할 수 있다.